Aggregate Distance Based Clustering Using Fibonacci Series-FIBCLUS

Rakesh Rawat, Richi Nayak, Yuefeng Li, Slah Alsaleh
Faculty of Science and Technology, Queensland University of Technology
Brisbane Australia
r.rawat,r.nayak,y2.li,s.alsaleh{@qut.edu.au}

Abstract. This paper proposes an innovative instance similarity based evaluation metric that reduces the search map for clustering to be performed. An aggregate global score is calculated for each instance using the novel idea of Fibonacci series. The use of Fibonacci numbers is able to separate the instances effectively and, in hence, the intra-cluster similarity is increased and the inter-cluster similarity is decreased during clustering. The proposed FIBCLUS algorithm is able to handle datasets with numerical, categorical and a mix of both types of attributes. Results obtained with FIBCLUS are compared with the results of existing algorithms such as k-means, x-means expected maximization and hierarchical algorithms that are widely used to cluster numeric, categorical and mix data types. Empirical analysis shows that FIBCLUS is able to produce better clustering solutions in terms of entropy, purity and F-score in comparison to the above described existing algorithms.

Keywords*:* Clustering numeric, categorical and mix datasets, Fibonacci series and golden ratio, similarity evaluation.

1. Introduction

Evaluation of similarity of attributes between instances is the core of any clustering method. The better a similarity function the better the clustering results would be. If the dataset contains numeric attributes, distance measures such as Euclidean, Manhattan and cosine, are effective to evaluate the similarity between objects [1],[2],[3]. However when the dataset contains categorical (finite and unordered) attributes or a mix of numeric and categorical attributes then such distance measures may not give good clustering results [3]. Comparison of a categorical attribute in two objects would either yield 1 for similar values and 0 indicating that two instances are dissimilar. Such similarity measures are defined as overlap measure [4], and mostly suffer from the problem of clustering dissimilar instances together when the number of attributes matched is same, but attributes that are matched are different [5]. Data driven similarity measures are becoming a focus of research [5]. Datasets containing a mix of numerical and categorical attributes have become increasingly common in modern real-world applications.

In this paper, we present a novel algorithm called as FIBCLUS (Fibonacci based Clustering) that introduces effective similarity measures for numeric, categorical and

a mix of both these types of attributes. Due to the mapping of all attributes of an instance to a global aggregate score, this method reduces the complexity inherent in the clustering process. Moreover, due to the use of Fibonacci numbers to separate the attribute values, this method enables higher intra-cluster similarity and lower inter-cluster similarity and, in hence, better clustering. Experiments with the proposed method are conducted using a total of 9 datasets, containing a mix of numeric, categorical and combinational attributes. The quality of clusters obtained is thoroughly analyzed. Empirical analysis shows that there was an average improvement of 14.6% in the purity values, 28.5% in the entropy values and about 8% in the F-score values of clusters obtained with FIBCLUS method on all the datasets in comparison to clustering solutions obtained using the existing methods such as k-means, x-means expected maximization and hierarchical algorithms .

The contributions of this paper can be summarized as: 1) A novel clustering similarity metrics that utilises Fibonacci series to find similarities between numerical, categorical and a mix of both the data types; 2) A global score representation method for these types of attributes; and 3) Enhancing existing clustering algorithms by using FIBCLUS as a similarity metrics.

2 Problem Statement

When pure categorical datasets or mixed datasets consisting of both the categorical and numerical attributes are to be clustered, the problem is how to measure the similarity between the instances represented by categorical attributes. A similarity measure, overlap, between two categorical instances X_i and X_j can be defined as follows:

$$S(X_i, X_j) \equiv \sum_{k=1}^{m} \delta(x_{ik}, x_{jk}), \text{ where } \delta(x_{ik}, x_{jk}) = \begin{Bmatrix} 1, & x_{ik} = x_{jk} \\ 0, & \mathbf{x}_{ik} \neq x_{jk} \end{Bmatrix}. \quad (1)$$

Such similarity measures may result in weak intra similarity when calculating the similarity between categorical attributes [2]. Other similarity measures for categorical attributes such as Eskin, Goodall, IOF, OF, Lin, Burnaby [5] are based on the overlap similarity measure and inherit the same problems. Moreover, in modern real-world applications, data with various instances containing a mix of both categorical and numerical attributes are common. A problem arises when assignment of an instance to a particular cluster is not easy. This problem is shown by the example in deck of cards problem.

Consider two datasets, one containing a single deck of 52 cards and another consisting of two decks of cards. Each deck of cards is identified by the distinct cover design it has. Clustering deck of cards may be a trivial problem, but it represents perfect clustering and the major shortcomings of clustering methods, which is when assignment of an instance to a cluster becomes difficult. As the number of deck increases, the number of clusters and the complexity inherent within the clustering process increases. As the number of deck increases from $1..n$ the number of perfect clusters increases to $4n$ where n is the number of decks. The ideal clustering results are shown in Table 2 for the deck of cards dataset problem. The corresponding

clustering results obtained by different algorithms such as expectation minimization (denoted as EM), K means (KM) and extended K means (XM) are shown in Table 3. These were implemented in Weka [6] with both Euclidian and Manhattan distances. Clustering using direct, repeated bisection and agglomerative were used with both the cosine and correlation coefficient similarity measures implemented in gcluto [1]. Only the best results observed are reported for all the methods.

Table1: Data description for deck of cards clustering problem.

SN	Attribute Name	Attribute type	Value Range	Description
1	Card No	Numeric/discrete	1-13	1-13 of all cards
2	Colour	Categorical	2	Red or Black
3	Category	Categorical	4	Hearts, Diamonds, Spade, Clubs
4	Deck Id	Numeric/Binary	1,2	1-1st Deck,2-2nd Deck

Table 2 Deck of cards cluster accuracy measure criteria (D1=deck1,D2=deck2).

2 Clusters	4 Clusters	8 Clusters
1-13, Red	1-13,Red , Hearts	1-13,Red , Hearts, D1
1-13, Black	1-13,Black , Spade	1-13,Red , Hearts, D2
	1-13,Black , Clubs	1-13,Red , Diamonds, D1
	1-13,Red, Diamonds	1-13,Red , Diamonds, D2
		1-13,Black , Spade, D1
		1-13, Black , Spade, D2
		1-13, Black , Clubs, D1
		1-13, Black , Clubs, D2

Table 3: Clustering results for decks of cards problem (D1=deck1,D2=deck2).

SN	Clustering Algorithm	Cluster=2 Correctly clustered		Cluster=4 Correctly Clustered		Cluster=8 Correctly clustered
		D1	D2	D1	D2	D2
1	EM	100%	100%	100%	100%	48.07%
2	KM	100%	98%	63.5%	62.5%	56.7%
3	XM	100%	98%	73.1%	62.5%	56.7%
4	Direct	25%	62.5%	38.5%	36.5%	31.7%
5	Repeated Bisection	25%	65.5%	48%	44.2%	31.8%
6	Agglomerative	48%	65.5%	33%	48%	25%
7	Clustering Functions #4, #5, #6 above with FIBCLUS	100%	100%	100%	100%	100%

Results clearly show that the mentioned clustering algorithms based on respective similarity measures perform satisfactory with a single deck of cards, but as the complexity increases the clustering performance starts decreasing (Table 3). This problem occurs due to the similarity methods adopted by such algorithms. Such methods are unable to handle the mix of attributes and their inherent relationships. As the number of deck increases from one to two, the distance measures or similarity methods employed by such methods start to overlap distances.

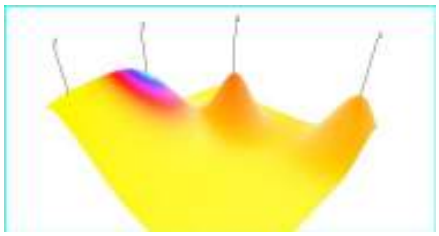

Fig1 (a) Agglomerative (4 Clusters)

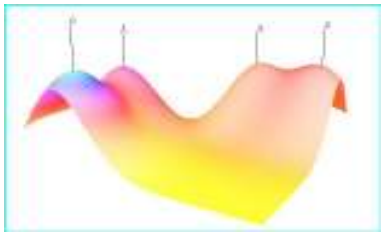

Fig1 (b) FIBCLUS with Agglomerative (4 Clusters)

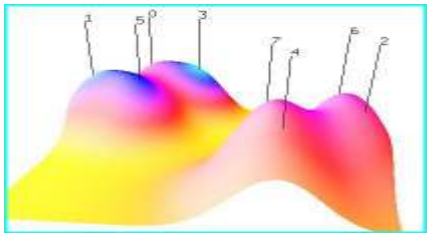

Fig1(c) Agglomerative (8 Clusters)

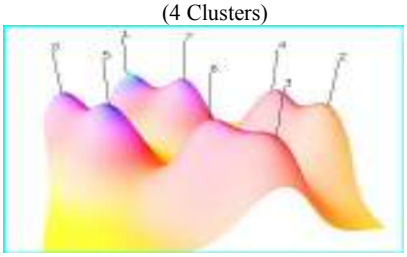

Fig1 (d) FIBCLUS with Agglomerative (8 Clusters)

The figures 1(a)-1(d) further visualize the cluster assignments for the 2 deck of cards. For agglomerative algorithm, the best of the cosine and correlation coefficient similarity measures was taken. With FIBCLUS and agglomerative clustering both measures gave same results. From figures 1(a), 1(c) it can be clearly deduced that clusters have overlapping distances, which consequently results in a weak clustering solution. The assignment of same peaks to a set of clusters shows the overlapping and, consequently a weak intra cluster similarity value. However in figures 1(b) and 1(d), with FIBCLUS, the clusters were clearly identifiable. The high peaks in figures 1(b) and 1(d) binding similar instances together confirm that the intra cluster similarity was maximized using the method, hence resulting in the desired and optimal clustering for the underlying problem. Further separate peaks for each of the 8 clusters reflects high inter cluster similarity.

3 Related work

K means clustering is one of the best known and commonly used algorithm. K means [7] were inherently developed to deal with numerical data, where distances between instances are a factor for clustering them together. The widely used distance measure functions adopted by K means are Euclidean, Manhattan and cosine. Several K means extensions have been developed to cluster categorical data [3],[7]. Authors in [7] developed an efficient algorithm which clusters categorical data using the K means concept. A dissimilarity function based on simple matching, which evaluates the dissimilarity between a categorical instance and the cluster representative is used. The frequencies of all attributes of the instance matching the cluster are used for calculating the dissimilarity. Another approach based on K means to cluster

categorical datasets [3] uses simple matching scheme, replaces means of clusters by modes and uses frequency to solve for the best clustering outputs.

A further classification of similarity evaluation for categorical data based on neighbourhood [8],[9],[10] and learning algorithms [11],[12] is discussed in [5]. Mostly neighbourhood based evaluation methods use similarity methods as adopted by the overlap measures [5]. Some of them are Eskin, Goodall, IOF, OF, Lin, Burnaby [2]. Unlike the overlap measure, these measures consider both similarity and dissimilarity between instances, assigning 1 for a perfect match and arbitrary small values for a mismatch. Rock [10] and Cactus [11] are some of the popular agglomerative hierarchical algorithms which are used for categorical data clustering. Rock clusters instances in an agglomerative way maximizing the number of links in a cluster whereas Cactus utilises co-occurrence of pairs of attributes values to summarise the data and to achieve linear scaling. Birch [13] and Coolcat [14] are other popular clustering methods used for clustering categorical data. Birch uses a balanced tree structure (CF tree) which preserves the attribute relationships within different instances as leaf nodes and then clustering is done on these leaf nodes. Coolcat is an incremental algorithm which achieves clustering by trying to minimize the entropy values between clusters. An approach [15] to cluster categorical and mix data uses a distance based similarity. A weighting scheme is adopted by the authors which utilizes the relative importance of each instance. Once distance between instances is evaluated a modified version of similarity metrics defined by [16] as $S(X_i,X_j)=1-d_p(X_i,X_j)$ is used to find instances similarities.

Simple similarity measures such as overlap suffer from the problem of clustering dissimilar instances together when the number of attributes matched is same, but attributes that are matched are different. Moreover, these similarity measures may perform well with categorical data, but in the case of mixed data which contains both numerical and categorical data the performance declines as the complexity within clusters increases.

4 The Fibonacci series and golden ratio

The proposed FIBCLUS (Fibonacci based Clustering) uses the Fibonacci series to determine a global score for each instance and then utilizes the aggregate distance as a similarity function. Fibonacci series is a sequence of numbers $\{F_n\}_{n=1}^{\infty}$ defined by the linear recurrence equation $F_n = F_{n-1} + F_{n-2}$. The first two Fibonacci numbers are 0 and 1, and each subsequent number is the sum of the previous two. The Fibonacci series has been applied in many scientific and real life fields [17] from analysis of financial markets, to development of computer algorithms such as the Fibonacci search technique and the Fibonacci heap data structure [18]. One of the prominent properties of Fibonacci series is that the ratio of two successive numbers F_n / F_{n-1}, where $n \geq 7$ tends towards 1.6 or φ, as n approaches infinity [17]. This value of φ is also called as the golden ratio.

The primary purpose of using Fibonacci series is, since each similar attribute of all instances are multiplied by a distinct successive Fibonacci number, only similar

attributes in different instances will have same values and will be clustered appropriately. If there are m categorical attributes in an instance which have been converted into equivalent numerical attributes then as we do Fibonacci transformation of the attribute from $1...m$ the ratio between $\frac{x_{i,2}}{x_{i,1}}, \frac{x_{i,3}}{x_{i,1}}, .. \frac{x_{i,m}}{x_{i,1}}$ will increase significantly, however for two successive attributes, it will always have a minimum values as φ. Due to this transformation property the ordering of attributes will have no major effect on the clustering solution, as the global scores per instance will be compared with each other when performing the clustering solution.

5 The proposed FIBCLUS method

The aim of using FIBCLUS with numeric data is to generate a search space in which the input instances are clearly distinguishable. FIBCLUS represents each instance as an aggregate global value compromising of various attributes. In other words, if there are n numeric instances and m number of attributes then the FIBCLUS reduces the search space for each $X = \{X_1, X_2, .., X_n\}$ from m to 1:

$$\mathbb{R}^n = \{(x_{n,1}, x_{n,2} .. x_{n,m})\} \rightarrow \{(x_{n,1})\} \tag{2}$$

For categorical and mix data the aim of FIBCLUS is to identify the best possible similarity that exists between a pair of instances by considering all the attributes. The score of all attributes in this case is also represented as an aggregate global score. Given the set of instances $X = \{X_1, X_2, .., X_n\}$ with m number of attributes $(x_{i,1}, x_{i,2}, ...x_{i,m})$, a Fibonacci number is initialized for each attribute maintaining the golden ratio φ. Let $F = \{F_1, F_2 ... F_m\}$ be the set of Fibonacci numbers chosen corresponding to m number of attributes where each successive Fibonacci number F_{j+1} maintains the golden ratio φ with the preceding number F_j. In the experiments F1 is initialized as $F_1 = 5$ because the series starts to get closer and closer to φ after this number. Consider an example for the dataset of four attributes $x_{i,1}, x_{i,2}, x_{i,3}, x_{i,4}$, where $F = \{5, 8, 13, 21\}$ is the set of Fibonacci numbers. In this case, $F_1 = 5$ is used to transform $x_{i,1}$ and $F_2 = 8$ is used to transform $x_{i,2}$ and so on. A value in F maintains the golden ratio as $F_2 / F_1, F_3 / F_2, F_4 / F_3 \cong 1.6$.

There are three cases, which have to be considered while clustering with FIBCLUS. **Case 1:** Clustering pure numeric attributes. In this case the maximum value of each attribute $\max(x_{i,1}), \max(x_{i,2}), ... \max(x_{i,m})$ is used for normalizing the attribute values. Normalization is done to scale the values in a constant range so that the Fibonacci number chosen for that attribute does not drastically change the golden ratio ϕ, which separates the values of one attribute from another. **Case 2:** For clustering pure categorical attributes each categorical attribute values are mapped into numeric values. Each instance X_i with attributes as $(x_{i,1}, x_{i,2}, ...x_{i,m})$, and Fibonacci

mapped value $F_j x_{i,j}$ is assigned a score. Each instance is compared for similarity with other instances. **Case 3:** In this case for clustering mix of both numeric and categorical attributes, let k be the number of categorical attributes, and l be the number of numeric attributes, where $k+l=m$. The score of each instance is determined separately based on the values of both numeric and categorical attributes (case 1 and case 2) as shown in step 3 of algorithm (figure 2).

$$Score(X_i) = \sum_{1}^{k} (x_{i,k} \times F_k) + \sum_{1}^{m} \frac{(x_{i,l})}{\max(x_{i,l})} \times F_l. \quad (3)$$

Input:

$X = \{X_1, X_2, .., X_n\}$;// Datasets instances with m attributes as $x_{i,1}, x_{i,2}, ..., x_{i,m}$.

$F = \{F_1, F_2 .. F_m\}$. //Successive Fibonacci numbers F corresponding to each $1..m$ attributes.

$F_j x_{i,j}$ = Categorical attribute values, mapped into numeric value .

Output:

$\mathbb{R}^n = \{(x_{n,1})\}$ // Numeric instances: Global score

$\mathbf{A} = [n \times n]$ // Categorical or Mix: Similarity Matrix.

Begin:

Step 1. $F_1 = 5$. // Initialize Fibonacci series.

Step 2. // For numeric attribute $\max(x_m)$ finds maximum attribute value from instances.

For each j=1 to m;

$\max(x_j)$

Step 3. // Evaluate scores for each instance.

For each i=1 to n;

$Score(X_i) = 0.0$;

For each j=1 to m;

If domain (x_{ij}) = Numeric

$$Score(X_i) = Score(X_i) + \frac{x_{i,j}}{\max(value_j)} \times F_j.$$

Else domain (x_{ij}) = Categorical

$$Score(X_i) = Score(X_i) + F_j x_{i,j}.$$

Step 4. //Calculate similarity between instances.

For each i=1..n;

For each j=1..n;

If ($(Score(X_i) <= Score(X_j))$

$$Similarity(X_i, X_j) = \frac{X_i \cap X_j}{m} + \frac{Score(X_i)}{Score(X_j)}$$

Return $\mathbb{R}^n = \{(x_{n,1})\}$ or $\mathbf{A} = [n \times n]$;

End.

Fig 2: Complete FIBCLUS Algorithm.

Finally, the instance similarity between two instances X_i, X_j is evaluated based on equation (3) as shown in equation (4) and figure 2 (Step 4), where $X_i \cap X_j$ is the number of similar categorical instances between the two instances and $Score(X_i) <= Score(X_j)$. This condition makes sure that the similarity calculation is done only once between pair of instances.

$$Similarity(X_i, X_j) = \frac{X_i \cap X_j}{m} + \frac{Score(X_i)}{Score(X_j)} \quad (4)$$

The pair wise similarity matrix between all instances denoted as $\mathbf{A} = [n \times n]$ becomes input to a clustering algorithm.

6 Empirical Analysis

The objective of experiments was to evaluate the quality of clustering results obtained using the proposed FIBCLUS similarity scores, adopted in the different clustering algorithms. Standard evaluation criteria such as Entropy, Purity and F-Score were used to assess the quality. For numeric datasets FIBCLUS was used with Expectation Minimization (EM), K means (KM) and Extended K means (XM) [6] shown as #1, #2 ,#3 respectively. For categorical and mix data we used direct, repeated bisection and agglomerative clustering methods implemented in gcluto [1] and shown as #1, #2 ,#3 in all results table(5,6,7). Correlation coefficient and cosine similarity were taken as similarity evaluation methods and the best results were taken. The test datasets were obtained from the UCI repository except Medical[1] as detailed in Table 4. A total of 9 datasets, three of each category were used in experiments. These datasets were taken due to clear class definitions of each instance, which could be compared accurately against results of various clustering methods.

Table 4: Clustering test datasets details.

SN	Dataset	Attribute Type	No. of Attribute	No. of class	No. of instance
1	Liver	Numeric	6	2	345
2	Wine	Numeric	13	3	178
3	IRIS	Numeric	4	3	150
4	Soybean	Categorical	35	4	47
5	Balance	Categorical	4	3	625
6	SpectHeart	Categorical	22	2	267
7	Teaching	Mix	5	3	151
8	Medical	Mix	8	3	90
9	Hepatitis	Mix	19	2	155

[1] Creators: Sharon Summers, School of Nursing, University of Kansas Medical Center, Kansas City, KS 66160,Linda Woolery, School of Nursing, University of Missouri, Columbia, MO 65211, Donor: Jerzy W. Grzymala-Busse (jerzy@cs.ukans.edu)

Overall as can be seen, the performance of all clustering algorithms improves when FIBCLUS based global scores and similarity scores are used. This happens due to the separation ratio that is actively bringing similar instances together (in hence making the intra-cluster similarity larger) and separating dissimilar instances more further from each other (in hence making the inter-cluster similarity lower). Independent of the type of attributes and the clustering process used, FIBCLUS is able to produce clustering solutions of high accuracy.

Table 5: Results of Purity of Clustering of all datasets.

	Purity of clustering results					
	Without FIB Values			**FIB Values With**		
Datasets	EM	KM	XM	#1	#2	#3
Liver	0.507	0.542	0.557	0.513	0.536	0.536
Wine	0.376	0.433	0.433	0.719	0.719	0.719
IRIS	0.907	0.887	0.880	0.960	0.960	0.960
Soybean	1.000	0.979	0.979	0.979	0.979	0.979
Balance	0.526	0.494	0.538	0.549	0.549	0.549
SpectHeart	0.528	0.614	0.614	0.772	0.772	0.772
Teaching	0.417	0.437	0.437	0.424	0.404	0.430
Medical	0.6	0.422	0.422	0.478	0.478	0.478
Hepatitis	0.516	0.542	0.542	0.775	0.763	0.755
Average	**0.597**	**0.594**	**0.6**	**0.685**	**0.684**	**0.686**

Table 6: Results of Entropy of Clustering of all datasets

	Entropy of clustering					
	Without FIB Values			**FIB Values With**		
Datasets	EM	KM	XM	#1	#2	#3
Liver	0.233	0.21	0.184	0.255	0.231	0.231
Wine	0.372	0.377	0.377	0.209	0.209	0.209
IRIS	0.103	0.128	0.141	0.05	0.05	0.05
Soybean	0	0.025	0.025	0.041	0.041	0.041
Balance	0.446	0.458	0.437	0.41	0.41	0.41
SpectHeart	0.195	0.188	0.188	0.012	0.012	0.012
Teaching	0.472	0.449	0.449	0.469	0.475	0.471
Medical	0.231	0.454	0.454	0.306	0.306	0.306
Hepatitis	0.300	0.297	0.297	0.01	0.017	0.021
Average	**0.261**	**0.287**	**0.284**	**0.196**	**0.195**	**0.195**

Table 7: Results of F-Score of Clustering of all datasets.

	F-Score of clustering					
	Without FIB Values			**FIB Values With**		
Dataset	EM	KM	XM	#1	#2	#3
Liver	0.438	0.459	0.457	0.461	0.467	0.469
Wine	0.336	0.376	0.376	0.713	0.713	0.713
IRIS	0.907	0.887	0.88	0.96	0.96	0.96
Soybean	1	0.985	0.985	0.975	0.975	0.975
Balance	0.456	0.437	0.484	0.448	0.448	0.448
SpecHeart	0.517	0.422	0.422	0.436	0.436	0.436
Teaching	0.420	0.433	0.433	0.425	0.408	0.433

Medical	0.333	0.301	0.301	0.332	0.332	0.332
Hepatitis	0.437	0.467	0.467	0.436	0.436	0.436
Average	**0.538**	**0.53**	**0.534**	**0.576**	**0.575**	**0.578**

When each cluster is visualized for its purity in figures 3(a)-3(f), the standard EM, KM and XM methods without any space mapping derives clusters with varied purity. For datasets like Iris and Soybean EM performed exceptionally well when compared to distance based algorithm like KM and XM. However when such datasets were used with FIBCLUS in general it was found out that the distance based algorithms like KM and XM performed much better than the density based algorithm like EM. This observation indicates that FIBCLUS has the ability to improve inter and intra cluster distances in any type of clustering method. For numeric datasets FIBCLUS works reasonably well. This is because the aggregate global score computed by FIBCLUS for each instance, is able to map various attributes to a greater extent. Since each attribute is well separated by the golden ratio, the overall score of similar instances is more similar. For some datasets like IRIS, unsupervised clustering using FIBCLUS is able to get 96% accuracy which is equal to some supervised learning methods like J48 [19]. This shows that the reduced search map obtained using the global score calculated using Fibonacci numbers is able to decrease the complexity of the grouping process. For the Wine dataset, results are exceptionally well. The performance improvement in clustering using FIBCLUS (#1, #2, #3) is nearly 50%. For the Liver dataset, results are nearly comparable, however the clustering achieved using it has better clusters which is evident from the purity and entropy measures.

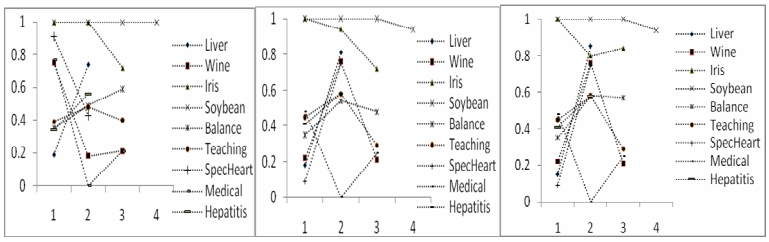

Fig3(a): Purity EM **Fig3(b):** Purity KM **Fig3(c):** Purity XM

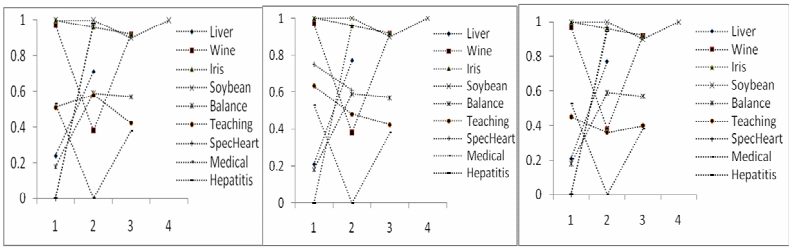

Fig3(d): Purity FIBCLUS(#1) **Fig3(e):** Purity FIBCLUS(#2) **Fig3(f):** Purity FIBCLUS(#3)

Overall the average results as percentage of various evaluation metrics are summarized in table 8.

Table 8: Summary of Results on test datasets.

Overall Percent(%) improvements in clustering using FIBCLUS		
Similarity Measure	**Best Case**-Best of all Clustering results Versus best FIBCLUS results is taken	**Worst Case**- Worst of all Clustering results Versus worst FIBCLUS result is taken
Purity	14%	15.15%
Entropy	25.29%	31.7%
F Score	7.4%	8.5%

7 Conclusion

This paper proposed an innovative clustering method that reduces the search map for clustering to be performed. An aggregate global score is calculated for each instance using the novel idea of Fibonacci series. Similarity functions are proposed by using the aggregate global score for instances with numerical, categorical or mix attributes. The use of Fibonacci numbers is able to separate the instances effectively and, in hence, enables a higher intra-cluster similarity and a lower inter-cluster similarity. The proposed FIBCLUS method is applied on a wide variety of datasets with categorical, numerical and mix attributes. FIBCLUS is compared with the existing algorithms that are widely used to cluster numeric, categorical and mix data types.

Empirical analysis shows that FIBCLUS is able to produce better clustering solutions in terms of entropy, purity, F-score etc in comparison to existing algorithms such as k-means, x-means, expected maximization and hierarchical algorithms. However the extra overhead in terms of time and space due to the additional step of calculating the similarity scores between instances in case of instances containing mix or categorical, is compensated by the reduced search map during the clustering process. Moreover, clustering usually is an offline process and is more affected by accuracy than such measures.

8 Acknowledgment

This research has been funded by CRC (Co-operative Research Centre), Australia and Queensland University of Technology, Brisbane Australia under the CRC Smart Services Project 2009-10.

References

1 Rasmussen, M. and Karypis, G., "gcluto: An interactive clustering, visualization, and analysis system." vol. 21: Citeseer, 2008.

2 Liao, H. and Ng, M. K., "Categorical data clustering with automatic selection of cluster number," *Fuzzy Information and Engineering,* vol. 1, pp. 5-25, 2009.

3 Huang, Z., "Extensions to the k-means algorithm for clustering large data sets with categorical values," *Data Mining and Knowledge Discovery,* vol. 2, pp. 283-304, 1998.

4 C. Stanfill., D. W., "Toward memory-based reasoning," *Communications of the ACM,* vol. 29, pp. 1213-1228, 1986.

5 Boriah, S., Chandola, V., and Kumar, V., "Similarity measures for categorical data: A comparative evaluation." vol. 30: Citeseer, 2007, p. 3.

6 Ian H. Witten, E. F., *Data Mining: Practical machine learning tools and techniques* vol. 2nd Edition: San Francisco: Morgan Kaufmann,, 2005.

7 San, O. M., Huynh, V. N., and Nakamori, Y., "An alternative extension of the k-means algorithm for clustering categorical data," in *Internation Journal of Applied Mathematics and Computer Science*. vol. 14: University of Zielona Gora Press, 2004, pp. 241-248.

8 Ahmad, A. and Dey, L., "A method to compute distance between two categorical values of same attribute in unsupervised learning for categorical data set," *Pattern Recognition Letters,* vol. 28, pp. 110-118, 2007.

9 S.Q.Le. and T. B. Ho., "An association-based dissimilarity measure for categorical data," *Pattern Recognition Letters,* vol. 26, pp. 2549-2557, 2005.

10 Guha, S., Rastogi, R., and Shim, K., "Rock: A robust clustering algorithm for categorical attributes* 1," *Information Systems,* vol. 25, pp. 345-366, 2000.

11 Ganti, V., Gehrke, J., and Ramakrishnan, R., "CACTUS—clustering categorical data using summaries," in *fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, San Diego, California, United States 1999, pp. 73-83.

12 Gibson, D., Kleinberg, J., and Raghavan, P., "Clustering categorical data: An approach based on dynamical systems," *The VLDB Journal,* vol. 8(3), pp. 222-236, 2000 2000.

13 Zhang, T., Ramakrishnan, R., and Livny, M., "BIRCH: an efficient data clustering method for very large databases," in *ACM SIGMOD, International Conference on Management of Data*, 1996, pp. 103-114.

14 Barbará, D., Li, Y., and Couto, J., "COOLCAT: an entropy-based algorithm for categorical clustering," in *11th International Conference on Information and knowledge Management* 2002, pp. 582-589.

15 Rendón, E. and Sánchez, J., "Clustering based on compressed data for categorical and mixed attributes." vol. 4109: Springer Berlin / Heidelberg, 2006, pp. 817-825.

16 Ichino, M., Yaguchi, H, "Generalized Minkoeski metrics for mixed feature-type data analysis," in *IEEE Transaction on Systems,Man and Cybernitics 24*, 1994, pp. 694–708.

17 Chandra, P. a. W., Eric W., "Fibonacci Number," in *MathWorld--A Wolfram Web Resource.http://mathworld.wolfram.com/FibonacciNumber.html*

18 Fredman, M. L. and Tarjan, R. E., "Fibonacci heaps and their uses in improved network optimization algorithms." vol. 34: ACM, 1987, pp. 596-615.

19 Lacueva-Pérez, F. J., "Supervised Classification Fuzzy Growing Hierarchical SOM," in *Hybrid Artificial Intelligence Systems: Third International Workshop, HAIS*, 2008, p. 220.